DOCUMENT RESUME

ED 280 856 TM 870 182

AUTHOR	Wise, Lauress L., II
TITLE	The PROC BRRVAR Procedure: Documentation. Technical Report No. 28.
INSTITUTION	American Institutes for Research in the Behavioral Sciences. Palo Alto, CA. Statistical Analysis Group in Education.
SPONS AGENCY	National Center for Education Statistics (ED), Washington, DC.
REPORT NO	AIR-87600-3-83-TR
PUB DATE	Mar 83
CONTRACT	300-78-0150
NOTE	48p.
PUB TYPE	Reports - Research/Technical (143) -- Computer Programs (101)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Computer Software; Elementary Secondary Education; *Error of Measurement; Estimation (Mathematics); Higher Education; National Surveys; *Sampling; Statistical Analysis; Statistical Bias
IDENTIFIERS	*Balanced Repeated Replication; *Sampling Error; Statistical Analysis System

ABSTRACT

BRRVAR, which uses the Balanced Repeated Replication approach, was designed for use with the Statistical Analysis System (SAS). It was created for the National Center for Education Statistics, to enlarge their capacity to estimate and analyze sampling errors for statistics generated from educational surveys with complex sampling designs. BRRVAR requires that the primary sampling units be organized into pairs that reflect the actual sampling design. Common statistics are generated, including sums, means, standard deviations, covariances, correlations, and regression coefficients. For each statistic, an overall estimate is computed along with the estimate of the standard error of the statistic and the approximate 95% confidence bounds for the statistic. The appendices include examples of an input dataset, SAS statements and output, and a computer program listing. (Author/GDC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED280856

Technical Report No. 28

The PROC BRRVAR Procedure: Documentation

Lauress L. Wise, II

Prepared by

SAGE
STATISTICAL ANALYSIS GROUP IN EDUCATION

For the

National Center for Education Statistics

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☒ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
W. V. Clemans
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

American Institutes for Research

Box 1113, Palo Alto, California 94302

TM 870 182

BEST COPY AVAILABLE

AIR-87600-3/83-TR

TECHNICAL REPORT 28

THE BRRVAR PROCEDURE: DOCUMENTATION

Submitted to the National Center for Education Statistics

by

Lauress L. Wise

Statistical Analysis Group in Education
American Institutes for Research
P. O. Box 1113
Palo Alto, California 94302

This work was done under Contract No. 300-78-0150 with the National Center for Education Statistics, Department of Health, Education, and Welfare. However, the content does not necessarily reflect the position or policy of either agency, and no official endoresment should be inferred.

March 1983

BEST COPY AVAILABLE

TABLE OF CONTENTS

Abstract 1

Introduction 3

Specification 4

PORC BRRVAR Statement 4

VAR Statement 6

STRATID Statement 6

PSUID Statement 6

WEIGHT Statement 7

Details 7

Example 8

References 9

Appendix A: Examples 11

Example: Input Dataset 13

Example: SAS Statements 14

Example: SAS Output 15

Appendix B: Program Listing 21

ABSTRACT

The BRRVAR procedure provides estimates of the sampling error or statistics generated from complex samples. BRRVAR uses the Balanced Repeated Replication approach (BRR), which requires that the primary sampling units be organized into pairs that reflect the actual sampling design. Common statistics are generated, including sums, means, standard deviations, covariances, correlations, and regression coefficients. For each statistic, an overall estimate is computed along with the estimate of the standard error of the statistic and the approximate 95% confidence bounds for the statistic.

INTRODUCTION

BRRVAR is a user-created procedure for use with the Statistical Analysis System (SAS). It was created for the National Center for Education Statistics to enlarge their capacity to estimate and analyze sampling errors for statistics generated from educational surveys with complex sampling designs. This program drew heavily from an earlier SAS procedure, NASSVAR, designed by David Morganstern and implemented by Greg Binzer, both of Westat, Inc. The most significant enhancements in the current program are the "automatic" application of an appropriate design matrix for defining the replicate half-samples and the much wider range of statistics which can be requested simply.

The use of BRRVAR may require considerable preparation of the input data file. If the initial sampling design was created specifically for BRR analyses, it specified sampling strata from which exactly two units were randomly drawn. (If the sampling was multi-stage, e.g. schools were sampled and then students were sampled within school, only the first or primary stage is of imporance here.) In this case, all that is required is that the file contain one or more variables that can by used to identify the different strata and the different PSUs within each stratum. The file must be sorted by these variables prior to calling BRRVAR.

In most cases, the actual sample drawn does not match the BRR assumptions exactly. It is then necessary to create pairs of primary sampling units and sort the file so that these pairs are adjacent. In some instances, the number of primary sampling units may be quite large. This is true, for example, if some primary sampling units were sampled with certainty so that the secondary units within these primary sampling units have to be treated as if they were the primary units. In such cases, it may be desirable to define "pseudo-PSUs" which are actually clusters of PSUs. If several schools were sampled with certainty for example, and then some number of students were sampled from each school, it would be desirable to define two randomly equivalent clusters of students for each school and then treat these clusters as "pseudo-PSUs". Similarly, if a relatively large number of schools were selected from some sampling stratum, it would be desirable to organize these schools into two randomly equivalent clusters and then use these clusters as "pleudo-PSUs."

In addition to specifiyng the variables that identify primary sampling units, it is necessary to specify a case weight using the WEIGHT statement. It is assumed that such a weight will have been put on the file for data from the kind of complex samples for which BRRVAR estimates are required. In general, the sampling weight will be the inverse of the overall sampling probability for each case (particularly if sums are of interest). If no such weight exists, then a dummy weight must be created.

Sepcification

The following statements are used with the BRRAVAR procedure:

PROC BRRVAR options and parms;

VAR(IABLES) variable list;

WEIGHT weight variable;

STRATID variable list;

PSUID variable list.

The PROC BRRVAR statement must include certain required parameters as described below. The VARIABLES, WEIGHT, STRATID, and PSUID statements are also required. Their function and syntax is described below.

PROC BRRVAR Statement

The following options may be provided in the PROC BRRVAR statement:

- DATA = <SAS dataset>, names the SAS dataset to be used by PROC BRRVAR. If no input dataset is specified, BRRVAR uses the most recently created SAS dataset.
- SUMS = causes sums to be estimated for each variable in the VARIABLE list.
- MEAN = causes means to be estimated for each variable in the VARIABLE list.

- STD = causes standard deviations to be estimated for each variable in the variable list.

- COV = causes covariances to be estimated for each variable in the variable list. (Note: this option is not compatible with the CORR option; if both are specified, correlations take precedence and covariances will not be printed.).

- CORR = causes correlations to be estimated for each pair of variables in the VARIABLE list.

- REGR = causes regression coefficients and a multiple R square to be estimated for each regresson equation specified. The last NDEP variables (1 if NDEP is not specified are treated as criterion variables. The remaining variables are treated as predictors. A separate regression equation is estimated for each criterion, using all of the predictor variables for each equation. If CORR has been specified, standardized regresssion coefficients are computed, otherwise raw regression coefficients are computed.

- NOMISS = causes listwise deletion of missing values (i.e., if any variable is missing, the entire case is excluded from all statistics). If only univariate statistics are requested and NOMISS is not specified, missing values will be deleted separately for each variable. If any of the multivariate statistics are requested (COV, CORR, or REGR) NOMISS is automatically forced.

- DESIGN = causes the replicate half-sample design matrix to be printed. The design matrix contains one row for each PSU and one column for each half-sample. The entries tell whether the PSU associated with the row was included in the half-sample indicated by the column.

- DUMP = requests a printout of the half-sample estimates for each statistic. In the current version this dump is relatively unformatted, with no labelling of the columns or rows.

- NPSU = <number of PSUs>, specifies the number of PSUs in the sample design. This parameter is primarily for documentation purposes at present. The actual number of PSUs is computed and compared to the expected number specified here. In case of discrepancies, the acutal number is used.

- NREP = <number of replicates>, specifies the number of half-sample replicates to be used. If omitted, NREP is set to the value of NPSU. (If both NPSU and NREP are

omitted, you are in some trouble.) In the current version, a maximum of 48 replicates can be created appropriately. If NREP is larger than this, the half-samples begin repeating the same design so that no new information is added.

- NDEP = <number of dependent variables>, this parameter must be omitted unless REGR is specified. It gives the number of dependent varialbes for which regressions are to be performed. The last NDEP variables in the variable list are taken as dependent variables. If REGR is specified, but NDEP is omitted, NDEP is set equal to 1.

VAR Statement

VAR variables;

The VAR statement lists the variables to be analyzed. All of the variables listed must be numeric. If omitted, all numeric variables in the input dataset will be analyzed.

STRATID Statement

STRATID variable(s);

The variables in the STRATID must be sufficient to uniquely identify the individual sampling stratum. The file must be sorted by the STRATID variables, otherwise execution will terminate. There should be exactly two PSUs within each sampling stratum. If only 1 is found, a "dummy" PSU with no cases is created for purposes of balancing the half-samples. If more than two PSU are found, the excess PSUs are ignored (after a warning message is printed).

PSUID Statement

PSUID variables;

The variables in the PSUID list must be sufficient to identify the individual PSUs within each stratum. The file must be sorted by the variables in the PSUID list within each stratum. As indicated above, there must be

exactly two PSUs within each sampling stratum. If the sample did not follow a strict BRR design, it may be necessary to construct "pseudo strata" or "pseudo PSUs" in order to obtain appropriate results. Consult your local sampling statistician.

WEIGHT Statement

WEIGHT variable;

The WEIGHT statement identifies the variable that is to be used as a weight for each case. One and only one weight variable must be specified. If the sample is unweighted, then a dummy weight variable must first be created. (If the sample is unweighted, do you really need a complicated algorithm like this?)

DETAILS

Missing Values

BRRVAR's treatment of missing values depends upon whether any multivariate statistics are requested. If COV, CORR, or REGR are specified, listwise deletion of cases with missing data is always performed. If only univariate statistics are requested, BRRVAR deletes missing values separately for each variable, unless NOMISS is specified in the PROC BRRVAR statement. If NOMISS is specified, then listwise deletion of cases with any missing values is always performed.

Missing values are not allowed for the weight variable. If any missing values are encountered, the program will terminate on the presumption that the weight variable was incorrectly specified.

Printed Output

The printed output from BRRVAR includes the following:

- a Summary Report indicating the number of cases processed and the number with missing values,

- a design matrix (if requested) indicating which PSUs were included in each of the replicate half-samples,

- all requested univariate statistics, including the overall estimate, the estimated sampling error of the overall estimate, and the lower and upper 95% confidence bounds for the overall estimate,

- if requested, an estimated correlation coeficient is printed for each pair of variables in the variable list along with the estimated sampling error of each correlation coefficient and the lower and upper confidence bounds of each correlation coefficient,

- if requested, BRRVAR prints a relatively unformatted dump of the estimates generated from each of the replicate half-samples. The columns of this dump correspond to the half-samples and the rows to the statistics being estimated. The first column gives the overall estimate.

EXAMPLE

The attached example in APPENDIX A shows the JCL necessary to access BRRVAR at COMNET. In this example, a single stratum variable and a single PSU identifier are read. The input dataset contains five records for each PSU with three variables for each record. The data are already sorted by PSU within stratum.

PROC BRRVAR is called twice. In the first instance, simple univarite statistics are requested. The second request asks for both correlations and a regression run. In addition, the DESIGN and DUMP options are invoked to provide additional information.

The printout resulting from the sample dataset is shown following the listing of the input records.

REFERENCES

Casady, R. J. The estimation of variance components using balanced repeated replications. Proceedings of the Social Statistics Section, American Statistical Association, 1975, 352-357.

Plackett, R. L. & Burman, P. J. The design of optimum multifactorial experiments. Biometrika, 1946, 33, 305-325.

APPENDIX A

Examples of an Input Dataset, SAS Statements and SAS Output

EXAMPLE: Input Dataset

```
00010 //X         JOB (ED,AIR),'SAGE-WYOUNG',PRTY=8,
00020 //              MSGCLASS=P,NOTIFY=XEJGNY,MSGLEVEL=(1,1)
00030 /*ROUTE PRINT R207
00040 // EXEC XSAS,OPTIONS='S=72',PRINT=P
00050 //WORK  DD UNIT=SYSDA,SPACE=(CYL,(5,5))
00060 //SASLIB DD DSN=NCES.XEJGNY.SAGELIB,DISP=SHR
00070 //SYSIN  DD *
00080 DATA TEMP;
00090       INPUT STR 1 PSU $2
00100             X    3-4  Y  5-6  Z  7-8;
00110
00120       WT=4.0;
00130 CARDS;
00140 1A 1 5 1
00150 1A 2 2 2
00160 1A 3 3 3
00170 1A 4 4 4
00180 1A 5 1 5
00190 1B 3 6 7
00200 1B 4 3 6
00210 1B 5 4 5
00220 1B 6 5 4
00230 1B 7 2 3
00240 2A 1 5 1
00250 2A 2 2 2
00260 2A 3 3 3
00270 2A 4 4 4
00280 2A 5 1 5
00290 2B 3 6 9
00300 2B 4 3 8
00310 2B 5 4 7
00320 2B 6 5 6
00330 2B 7 2 5
00340 3A 1 3 3
00350 3A 2 9 4
00360 3A 3 8 5
00370 3A 4 1 6
00380 3A 5 4 7
00390 3B 3 7 5
00400 3B 4 3 4
00410 3B 5 4 3
00420 3B 6 1 2
00430 3B 7 3 1
00440 4A 1 9 5
00450 4A 212 6
00460 4A 314 7
00470 4A 418 8
00480 4A 516 9
00490 4B 313 7
00500 4B 412 6
00510 4B 519 5
00520 4B 621 4
00530 4B 716 3
00540 ;
```

EXAMPLE: SAS Statements

```
00550
00560 PROC BRRVAR NREP=8 SUM MEAN STD;
00570      VAR X Y Z;
00580      STRATID STR;
00590      PSUID PSU;
00600      WEIGHT WT;
00610 ;
00620 PROC BRRVAR NPSU=9 NREP=8 MEAN SUM STD REGR NDEP=1 DESI
GN DUMP;
00630      VAR X Y Z;
00640      STRATID STR;
00650      PSUID PSU;
00660      WEIGHT WT;
00670 PROC CORR;
00680      VAR X Y Z;
```

EXAMPLE: SAS Output

S T A T I S T I C A L A N A L Y S I S S Y S T E M 3:30 THURSDAY, MARCH 10, 1983 1
PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
SUMMARY INFORMATION

40 OBSERVATIONS READ
160 WEIGHTED OBSERVATIONS READ

MISSING VALUES DELETED SEPARATELY FOR EACH VAR

9 REPLICATES IN 8 PSU PAIR DESIGN

S T A T I S T I C A L A N A L Y S I S S Y S T E M 3:30 THURSDAY, MARCH 10, 1983 2
PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
UNIVARIATE STATISTICS

					95% CONFIDENCE BOUNDS		
STATISTIC	VARIABLE	WTD. N	ESTIMATE	STD. ERROR	LOWER	UPPER	LABEL
SUM	X	160.0	640.000	80.000	483.200	796.800	
SUM	Y	160.0	1052.000	62.354	929.786	1174.214	
SUM	Z	160.0	760.000	105.830	552.573	967.427	
MEAN	X	160.0	4.000	0.500	3.020	4.980	
MEAN	Y	160.0	6.575	0.390	5.811	7.339	
MEAN	Z	160.0	4.750	0.661	3.454	6.046	
STD	X	160.0	1.732	0.124	1.489	1.975	
STD	Y	160.0	5.449	0.505	4.460	6.439	
STD	Z	160.0	2.107	0.271	1.575	2.638	

EXAMPLE: SAS Output (cont.)

```
                    S T A T I S T I C A L   A N A L Y S I S   S Y S T E M      3:30 THURSDAY, MARCH 10, 1983   3
                          PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                          HALF SAMPLE DESIGN MATRIX

PSU              MATRIX

     1 A       0 1 0 1 0 0 1 1
     1 B       1 0 1 0 1 1 0 0
     2 A       0 0 1 0 0 1 1 1
     2 B       1 1 0 1 1 0 0 0
     3 A       0 1 0 0 1 1 1 0
     3 B       1 0 1 1 0 0 0 1
     4 A       0 0 0 1 1 1 0 1
     4 B       1 1 1 0 0 0 1 0
       9 PSUS SPECIFIED
       8 PSUS FOUND


                    S T A T I S T I C A L   A N A L Y S I S   S Y S T E M      3:30 THURSDAY, MARCH 10, 1983   4
                          PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES
                          SUMMARY INFORMATION


        40 OBSERVATIONS READ
       160 WEIGHTED OBSERVATIONS READ

         0 WEIGHTED OBS DELETED FOR MISSING VALUES

         9  REPLICATES IN   8 PSU PAIR DESIGN
```

EXAMPLE: SAS Output (cont.)

STATISTICAL ANALYSIS SYSTEM 3:30 THURSDAY, MARCH 10, 1983 5

PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES

UNIVARIATE STATISTICS

STATISTIC	VARIABLE	WTD. N	ESTIMATE	STD. ERROR	95% CONFIDENCE BOUNDS LOWER	UPPER	LABEL
SUM	X	160.0	640.000	80.000	483.200	796.800	
SUM	Y	160.0	1052.000	62.354	929.786	1174.214	
SUM	Z	160.0	760.000	105.830	552.573	967.427	
MEAN	X	160.0	4.000	0.500	3.020	4.980	
MEAN	Y	160.0	6.575	0.390	5.811	7.339	
MEAN	Z	160.0	4.750	0.661	3.454	6.046	
STD	X	160.0	1.732	0.124	1.489	1.975	
STD	Y	160.0	5.449	0.505	4.460	6.439	
STD	Z	160.0	2.107	0.271	1.575	2.638	

STATISTICAL ANALYSIS SYSTEM 3:30 THURSDAY, MARCH 10, 1983 6

PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES

CORRELATION ESTIMATES

FIRST VARIABLE	SECOND VARIABLE	CORRELATION WTD. N	ESTIMATE	STD. ERROR	95% CONFIDENCE BOUNDS LOWER	UPPER	PROB C>0
Y	X	160.0	0.026	0.260	-0.482	0.535	
Z	X	160.0	0.069	0.351	-0.618	0.756	
Z	Y	160.0	0.341	0.273	-0.194	0.877	

EXAMPLE: SAS Output (cont.)

STATISTICAL ANALYSIS SYSTEM 3:30 THURSDAY, MARCH 10, 1983

PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES

REGRESSION COEFFICIENTS

FIRST VARIABLE	SECOND VARIABLE	REGRESSION WTD. N	ESTIMATE	STD. ERROR	95% CONFIDENCE BOUNDS LOWER	UPPER	PROB C>0
Z	X	160.0	0.033	0.410	-0.770	0.836	
Z	Y	160.0	0.342	0.284	-0.216	0.899	
MULT RSQ	Z	160.0	0.127	0.252	-0.366	0.621	

STATISTICAL ANALYSIS SYSTEM 3:30 THURSDAY, MARCH 10, 1983

PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES

REPLICATE STATISTICS

FIRST VARIABLE	SECOND VARIABLE	REPLICATE S WTD. N	ESTIMATE	STD. ERROR	95% CONFIDENCE BOUNDS LOWER	UPPER	PROB C>0
0.033	-0.662	0.042	-0.371	0.438	0.432	0.524	0.043	0.395
0.342	0.048	-0.051	0.314	0.639	0.471	0.631	0.260	0.800
0.127	0.442	-0.010	0.248	0.277	0.147	0.496	0.173	0.590
160.000	160.000	160.000	160.000	160.000	160.000	160.000	160.000	160.000
1.732	1.414	1.732	1.658	1.732	1.732	1.658	1.658	1.658
5.449	5.783	5.886	5.925	4.928	4.766	4.924	6.030	5.052
2.107	2.000	2.000	1.732	2.449	1.732	2.000	1.732	2.236
160.000	160.000	160.000	160.000	160.000	160.000	160.000	160.000	160.000
0.026	-0.043	0.284	0.153	-0.281	-0.315	-0.138	0.455	-0.146
0.069	-0.707	0.289	-0.174	0.000	0.000	0.302	0.522	0.135
0.341	0.048	-0.047	0.307	0.588	0.424	0.619	0.206	0.783
640.000	800.000	640.000	720.000	640.000	640.000	560.000	560.000	560.000
1052.000	1112.000	1128.000	1072.000	976.000	1072.000	1032.000	1088.000	936.000
760.000	800.000	800.000	640.000	800.000	960.000	800.000	640.000	640.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

EXAMPLE: SAS Output (cont.)

VARIABLE	N	MEAN	STD DEV	SUM	MINIMUM	MAXIMUM
X	40	4.00000000	1.75411604	160.00000000	1.00000000	7.00000000
Y	40	6.57500000	5.51867459	263.00000000	1.00000000	21.00000000
Z	40	4.75000000	2.13337340	190.00000000	1.00000000	9.00000000

CORRELATION COEFFICIENTS / PROB > "R" UNDER HO:RHO=0 / N = 40

	X	Y	Z
X	1.00000	0.02649	0.06852
	0.0000	0.8711	0.6744
Y	0.02649	1.00000	0.34138
	0.8711	0.0000	0.0311
Z	0.06852	0.34138	1.00000
	0.6744	0.0311	0.0000

APPENDIX B

Program Listing

```
00010 //X        JOB (ED.AIR).'D-WYOUNG'.PRTY=8.
00020 //             MSGCLASS=P.NOTIFY=XEJ0NY.MSGLEVEL=(1.1)
00030 /*ROUTE PRINT R207
00040 //WYYSTEP1 EXEC XPLIXCL.SYSOUT=P.
00050 //   PARM.PLI='NM.ND.A.NEST.X.NC(S).MAR(2.72.1).NSYN(C).NOP.OF.C'.
00060 //   PARM.LKED='MAP.XREF.LIST'.
00070 //   COND.LKED=(4.LT.PLI)
00080 //PLI.SYSIN  DD *
00090     MAIN:PROC OPTIONS(MAIN) :
00100          /*-------------------------------------------------*/
00110          /*                                                 */
00120          /*   BRRVAR : SAMPLING VARIANCE ESTIMATES USING    */
00130          /*            BALANCED REPEATED REPLICATIONS (BRR):*/
00140          /*                                                 */
00150          /*   PROC BRRVAR  < OPTIONS PARMS >:               */
00160          /*        VAR     < VARIABLE LIST >:               */
00170          /*        STRATID <STRATUM IDENTIFIER(S) >:        */
00180          /*        PSUID   < PSU IDENTIFIER(S) >:           */
00190          /*                NOTE: FILE MUST BE SORTED BY     */
00200          /*                      STRATUM AND PSU IDENTIFIERS:*/
00210          /*        WEIGHT  < WEIGHT VARIABLE >:             */
00220          /*                NOTE: WEIGHT VARIABLE LIST MUST  */
00230          /*                CONTAIN EXACTLY 1 VARIABLE       */
00240          /*        ADJBY   < ADJ FILE IDENTIFIERS >:        */
00250          /*                - NOT YET IMPLEMENTED -          */
00260          /*                MUST BE A SUBSET OF THE PSU      */
00270          /*                IDENTIFIERS:                     */
00280          /*        COMPUTE < COMPUTED VARIABLES >:          */
00290          /*                - NOT YET FULLY IMPLEMENTED -    */
00300          /*                                                 */
00310          /*   OPTIONS:                                      */
00320          /*                                                 */
00330          /*      DESIGN-PRINT DESIGN MATRIX FOR HALF SAMPLES*/
00340          /*        DUMP-PRINT STATISTICS FOR EACH HALF SAMPLE*/
00350          /*         SUM-ESTIMATE SUM OF EACH VARIABLE       */
00360          /*        MEAN-ESTIMATE MEAN OF EACH VARIABLE      */
00370          /*        STD -ESTIMATE  STD OF EACH VARIABLE      */
00380          /*        COV -ESTIMATE  COV MATRIX FOR THE VARIABLES*/
00390          /*        CORR-ESTIMATE  COR MATRIX FOR THE VARIABLES*/
00400          /*        REGR-PERFORM REGRESSION                  */
00410          /*             NOTE: (1) LAST VARIABLE IN VAR LIST */
00420          /*                       IS THE DEPENDENT VARIABLE */
00430          /*                   (2) IF CORR IS SPECIFIED.     */
00440          /*                       STANDARDIZED BETAS ARE    */
00450          /*                       ESTIMATED. OTHERWISE RAW  */
00460          /*                       REGRESSION COEFFICIENTS ARE*/
00470          /*                       SHOWN:                    */
00480          /*                                                 */
00490          /*   PARAMETERS:                                   */
00500          /*                                                 */
00510          /*        NPSU-TOTAL NUMBER OF PSU'S IN THE SAMPLE */
00520          /*                                                 */
00530          /*        NREP-(OPTIONAL) NUMBER OF REPLICATE HALF-*/
00540          /*             SAMPLES TO BE USED                  */
00550          /*             NOTE: IF OMITTED NREP IS            */
00560          /*                   MIN(100.NPSU/2)               */
00570          /*        NDEP-NUMBER OF DEPENDENT VARIABLES IF    */
00580          /*             REGRESSION IS SPECIFIED             */
00590          /*             NOTE: THE LAST NDEP VARIABLES IN THE*/
```

```
00600          /*                    VAR LIST ARE TAKEN AS DEPENDENT */
00610          /*                    VARIABLES.  EACH IS REGRESSED   */
00620          /*                    AGAINST THE PREDICTOR (NON-     */
00630          /*                    DEPENDENT) VARIABLES.           */
00640          /*                    DEFAULT=1.                      */
00650          /*                                                    */
00660          /*   EXTERNAL ROUTINES:                               */
00670          /*                                                    */
00680          /*        SETPARM                                     */
00690          /*        ALLOC                                       */
00700          /*        PROCESS                                     */
00710          /*        PRINTIT                                     */
00720          /*                                                    */
00730          /*                                                    */
00740          /*----------------------------------------------------*/
00750          /*                                                    */
00760          /*   DECLARE LINKAGE TO EXTERNAL SAS ROUTINES         */
00770          /*                                                    */
00780          DCL SASPLO ENTRY.
00790              UNUSED ENTRY(FIXED BIN(31)).
00800              BYPASS ENTRY(FIXED BIN (31)).
00810              SASLOG ENTRY(FIXED BIN(31),FIXED BIN(31)) :
00820          /*                                                    */
00830          /*   DECLARE EXTERNAL ROUTINES..                      */
00840          /*                                                    */
00850          DCL SETPARM ENTRY((*,*)FLOAT BIN(53)).
00860              ALLOC ENTRY((*,*)FLOAT BIN(53)).
00870               PRINTIT ENTRY((*,*) FLOAT BIN(53)).
00880              PROCESS ENTRY((*,*)FLOAT BIN(53)) :
00890          /*                                                    */
00900          /*   DECLARE EXTERNAL PROGRAM VARS..                  */
00910          /*                                                    */
00920          DCL (MEANFLG.
00930               STDFLG.
00940               COVFLG.
00950               CORRFLG.
00960               REGRFLG.
00970               SUMFLG.
00980               SSQFLG.
00990               NOMSFLG.
01000               DESFLG.
01010               DUMFLG.
01020               OUTFLG) BIT(1) EXTERNAL.
01030              (INPTR.
01040               PSUPTR.
01050               PSUPTR2.
01060               STRPTR.
01070               STRPTR2.
01080               OUTPTR) POINTER EXTERNAL.
01090              (NREP.
01100               NV1.
01110               NSTAT.
01120               NCOMP.
01130               NDEP.
01140               NPSUID.
01150               NSTRID.
01160               NADJID.
01170               NOBS.
01180               NMISS.NTOT) FIXED BIN(31) EXTERNAL.
01190               (WGTOBS.WGTMISS.NUMPSU) FLOAT BIN(53) EXTERNAL.
```

```
01200              ESTAB(1.1) FLOAT BIN(53) CONTROLLED :
01210          DCL I FIXED BIN(31) STATIC:
01220          /*                                                    */
01230          /*   MESSAGES...                                      */
01240          /*                                                    */
01250            DCL MSG1  CHAR(80) INIT(' VERSION 2A').
01260              MSG1A CHAR(80) INIT(' PREPARED FOR N.C.E.S.').
01270              MSG1B CHAR(80) INIT(' BY THE STATISTICAL ANANLSYS').
01280              MSG1C CHAR(80) INIT('    GROUP IN EDUCATION (SAGE)').
01290              MSG2 CHAR(80) INIT(' AMERICAN INSTITUTES FOR RESEARCH
   ').
01300              MSG3 CHAR(80) INIT(' P. O. BOX 1113').
01310              MSG4 CHAR(80) INIT(' PALO ALTO, CA 94302').
01320              MSG5 CHAR(80) INIT(' (415) 493-3550').
01330              MSGF1 FIXED BIN(31) BASED(ADDR(MSG1)).
01340              MSGF1A FIXED BIN(31) BASED(ADDR(MSG1A)).
01350              MSGF1B FIXED BIN(31) BASED(ADDR(MSG1B)).
01360              MSGF1C FIXED BIN(31) BASED(ADDR(MSG1C)).
01370              MSGF2 FIXED BIN(31) BASED(ADDR(MSG2)),
01380              MSGF3 FIXED BIN(31) BASED(ADDR(MSG3)).
01390              MSGF4 FIXED BIN(31) BASED(ADDR(MSG4)),
01400              MSGF5 FIXED BIN(31) BASED(ADDR(MSG5)):
01410          /*                                                    */
01420          /*   CREATE SAS ENVIRONMENT.. THEN ROCK & ROLL..      */
01430          /*                                                    */
01440          CALL SASPLO :
01450          ALLOCATE ESTAB :
01460          CALL SETPARM(ESTAB):
01470          CALL ALLOC(ESTAB):
01480          CALL PROCESS(ESTAB):
01490          CALL PRINTIT(ESTAB);
01500          /*                                                    */
01510          /*   PRINT MESSAGES AND HALT..                        */
01520          /*                                                    */
01530          CALL BYPASS(I):
01540          IF I=0 THEN DO:
01550             CALL UNUSED(-1) :            /*   PRINT CORE MSG..   *
   /
01560             CALL SASLOG(MSGF1.80) :
01570             CALL SASLOG(MSGF1A.80) :
01580             CALL SASLOG(MSGF1B.80) :
01590             CALL SASLOG(MSGF1C.80) :
01600             CALL SASLOG(MSGF2.80) :
01610             CALL SASLOG(MSGF3.80) :
01620             CALL SASLOG(MSGF4.80) :
01630             CALL SASLOG(MSGF5.80) :
01640             CALL SASLOG(MSGF1.80) :
01650          END:
01660    END MAIN :
01670   /*MEM2*/
01680 /*
01690 //LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=OLD
01700 //LKED.SYSLIB DD DSN=SYS2.PLIXLINK.DISP=SHR
01710 //             DD DSN=SYS2.PLIXBASE.DISP=SHR
01720 //             DD DSN=SYS2.PLIXTASK.DISP=SHR
01730 //             DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
01740 //             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
01750 //LKED.SYSIN  DD *
01760   ENTRY  ENTRY
01770   INCLUDE SYSLIB(SASPLO,PROCESS,ALLOC,SETPARM)
```

```
01780  NAME BRRVAR2(R)
01790 /*
END OF DATA
```

```
00010 //X        JOB (ED.AIR),'D-WYOUNG',PRTY=8,
00020 //             MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
00030 /*ROUTE PRINT R207
00040 //WYYSTEP1 EXEC XPLIXCL,SYSOUT=F,
00050 //   PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,OF,C',
00060 //   PARM.LKED='MAP,XREF,LIST,NCAL',
00070 //   COND.LKED=(4,LT,PLI)
00080 //PLI.SYSIN  DD *
00090   ALLOC:PROC(ESTAB) :
00100         /*-------------------------------------------------------*/
00110         /*                                                       */
00120         /*  ALLOC-DETERMINES AMOUNT OF CORE REQUIRED FOR         */
00130         /*         NASSVAR DATA STRUCTURES..                     */
00140         /*                                                       */
00150         /*         ALLOCATES REQUIRED CORE                       */
00160         /*                                                       */
00170         /*         ZEROS CORE ALLOCATED                          */
00180         /*                                                       */
00190         /*         EXTERNAL FUNCTION 'ALLOTRN' IS INVOKED        */
00200         /*         TO DETERMINE IF COMPVAR AND OUTVAR            */
00210         /*         STATEMENTS ARE USED..  'ALLOTRN'              */
00220         /*         TAKES CARE OF ALLOCATING CORE FOR THESE       */
00230         /*         STRUCTURES.. 'ALLOTRN' RETURNS '1'B IF        */
00240         /*         THE COMPVAR/OUTVAR CONSTRUCTION IS USED..     */
00250         /*                                                       */
00260         /*-------------------------------------------------------*/
00270         /*                                                       */
00280         /*   DECLARE EXTERNAL LINKAGE TO SAS..                   */
00290         /*                                                       */
00300         DCL NOVAR ENTRY(FIXED BIN(31))
00310                   RETURNS(FIXED BIN(31)),
00320             GETMEM ENTRY(FIXED BIN(31),FIXED BIN(31),PTR,
00330                          FIXED BIN(31)),
00340             ZERO ENTRY(PTR,FIXED BIN(31)),
00350             MEMERR ENTRY(FIXED BIN(31)),
00360             LODOUT ENTRY;
00370         /*                                                       */
00380         /*   EXTERNAL VARIABLES...                               */
00390         /*                                                       */
00400         DCL (INPTR,PSUPTR,PSUPTR2,STRPTR,STRPTR2,OUTPTR)
00410             PTR EXTERNAL,
00420             (COVFLG,CORRFLG,REGRFLG,OUTFLG,SSQFLG,NOMSFLG)
00430             BIT(1) EXTERNAL,
00440             (MEANFLG,STDFLG,SUMFLG) BIT(1) EXTERNAL,
00450             (NSTAT,NREP,NV1,NPSUID,NADJID,NSTRID)
00460             FIXED BIN(31) EXTERNAL,
00470              ESTAB(*,*) FLOAT BIN(53) CONTROLLED :
00480         /*                                                       */
00490         /*   LOCALS..                                            */
00500         /*                                                       */
00510         DCL (NV,NV2,NREPL,NEED,GOT) FIXED BIN(31),
00520              WORKVEC(1) FLOAT BIN(53) BASED(INPTR),
00530              PTR PTR,
00540              NULL BUILTIN;
00550         /*                                                       */
00560         /*  FREE ESTIMATE TAB... PATCH TO RUN UNDER THE          */
00570         /*  OPTIMIZER WHICH REQUIRES THAT A CONTROLLED           */
00580         /*  STRUCTURE MUST BE ALLOCATED IN ORDER TO BE           */
00590         /*  PASSED.. (NOT THE CASE WITH THE 'F' COMPILER..       */
```

```
00600          /*  AT THIS POINT WE HAVE NO IDEA WHAT DIMENSIONS      */
00610          /*  THIS TABLE MUST HAVE.... WILL ALLOCATE IT LATER    */
00620          /*  IN THIS MODULE....                                 */
00630          /*                                                     */
00640          FREE ESTAB :
00650          /*   LOAD SAS OUTPUT MODS IF OUTPUT SPECIFIED BEFORE   */
00660          /*   ALLOCATING ANY CORE(OUTMODS NEED ABOUT 1K...)     */
00670          /*                                                     */
00680           OUTFLG='0'B:   /* NOT YET IMPLEMENTED */
00690          IF OUTFLG THEN CALL LODOUT :
00700          /*                                                   */
00710          /*     NEED=DOUBLE WORDS REQUIRED FOR TEMPORY        */
00720          /*          STORAGE                                  */
00730          /*                                                   */
00740          /*         NV1-FOR INPUT VARS                        */
00750          /*       NPSUID-FOR PSU ID VARS                      */
00760          /*       NPSUID-FOR PRIOR PSU ID                     */
00770          /*       NSTRID-FOR STRAT ID VARS                    */
00780          /*       NSTRID-FOR PRIOR STRAT ID                   */
00790          /*                                                   */
00800           NEED=8*NV1+16*NPSUID+16*NADJID+16*NV1:
00810           CALL GETMEM(NEED,GOT,INPTR,0):
00820           IF NEED < GOT THEN DO:
00830              CALL MEMERR(0):
00840              STOP:
00850           END:
00860           PSUPTR=ADDR(WORKVEC(NV1+1)):
00870           PSUPTR2=ADDR(WORKVEC(NV1+NPSUID+1)):
00880           STRPTR=ADDR(WORKVEC(NV1+2*NPSUID+1)):
00890           STRPTR2=ADDR(WORKVEC(NV1+2*NPSUID+NSTRID+1)):
00900
00910          /*                                                     */
00920          /*   ZERO REGION RETURNED...                           */
00930          /*                                                     */
00940          CALL ZERO(INPTR,NEED) :
00950          /*                                                   */
00960          /*    NSTAT=DOUBLE WORDS REQUIRED FOR EACH           */
00970          /*          REPLICATE HALF SAMPLE:                   */
00980          /*                NV1=NV+1                           */
00990          /*                NV2=NV*NV1/2                       */
01000          /*           NV1 - FOR RECORD COUNTS (IF \NOMISS)    */
01010          /*           NV1 - FOR WEIGHTED SUMS (IF \NOMISS)    */
01020          /*     NV1+NCOMP - FOR SUMS/MEANS                    */
01030          /*           NV1 - FOR SSQS/STDS                     */
01040          /*           NV2 - FOR SSCP/COV/CORR                 */
01050          /*                                                   */
01060          NV=NV1-1:
01070          IF COVFLG ^  CORRFLG ^  REGRFLG THEN DO:
01080             SSQFLG='1'B:
01090             NV2=NV1*NV/2:
01100          END:
01110          ELSE DO:
01120             SSQFLG='0'B:
01130             NV2=0:
01140          END:
01150          NSTAT=2*NV1+NV2:
01160          IF NOMSFLG THEN NSTAT=NSTAT+4:
01170                     ELSE NSTAT=NSTAT+2*NV1:
01180          OUTPTR=NULL:
01190          /*         OUTPUT NOT SUPPORTED                      */
```

```
01200           /*                                                    */
01210           /*   ALLOCATE 2 DIM TABLE TO HOLD ESTIMATES..         */
01220           /*                                                    */
01230           NREP1=NREP+1:
01240           NEED=8*NREP1*NSTAT:
01250           CALL GETMEM(-1,GOT,PTR,0) :
01260           IF NEED > GOT THEN DO :
01270              CALL MEMERR(NEED-GOT) :
01280              STOP :
01290           END :
01300           ALLOCATE ESTAB(NREP1,NSTAT):
01310    END ALLOC :
01320   /*MEM4*/
01330 /*
01340 //LKED.SYSLMOD DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
01350 //LKED.SYSLIB DD
01360 //             DD
01370 //             DD
01380 //             DD
01390 //             DD
01400 //             DD DSN=NCES.XEJQNY.SAGELIB.DISP=SHR
01410 //             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
01420 //             DD DSN=SYS2.SAS.R795.GLIBRARY.DISP=SHR
01430 //             DD DSN=SYS2.PLIXBASE.DISP=SHR
01440 //LKED.SYSIN  DD *
01450  NAME ALLOC(R)
01460 /*
END OF DATA
```

```
00010 //X        JOB (ED,AIR),'D-WYOUNG',PRTY=8,
00020 //             MSGCLASS=P,NOTIFY=XEJGNY,MSGLEVEL=(1,1)
00030 /*ROUTE PRINT R207
00040 //WYYSTEP1 EXEC XPLIXCL,SYSOUT=P,
00050 //   PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,DF,C',
00060 //   PARM.LKED='MAP,XREF,LIST,NCAL',
00070 //   COND.LKED=(4,LT,PLI)
00080 //PLI.SYSIN  DD *
00090   SETPARM:PROC(ESTAB) :
00100         /*-------------------------------------------------------*/
00110         /*                                                       */
00120         /*.  SETPARM-DISCOVERS PROCEDURE OPTIONS AND PARMS      */
00130         /*         VALIDATES SAMPLE DESIGN..                     */
00140         /*                                                       */
00150         /*-------------------------------------------------------*/
00160         /*                                                       */
00170         /*   DECLARE EXTERNAL LINKAGES...                        */
00180         /*                                                       */
00190         DCL IOPT ENTRY(FIXED BIN(31))
00200                  RETURNS(FIXED BIN(31)).
00210             PARM ENTRY(FIXED BIN(31)) RETURNS(FLOAT BIN(53)).
00220             ERROR ENTRY(FIXED BIN(31),(*,*) FLOAT BIN(53)).
00230             OBSERR ENTRY.
00240             NOVAR ENTRY(FIXED BIN(31)) RETURNS(FIXED BIN(31)) :
00250         /*                                                       */
00260         /*   DECLARE EXTERNAL PROGRAM VARIABLES...               */
00270         /*                                                       */
00280         DCL (NREP,NV1,NDEP,NPSUID,NSTRID,NADJID,NCOMP)
00290             FIXED BIN(31) EXTERNAL,
00300              NUMPSU FLOAT BIN(53) EXTERNAL,
00310              ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
00320             (SSQFLG,CORRFLG,MEANFLG,STDFLG,COVFLG,REGRFLG,SUMFLG,
00330              DESFLG,DUMFLG,NOMSFLG) BIT(1) EXTERNAL;
00340         /*                                                       */
00350         /*   ERROR MESSAGES..                                    */
00360         /*                                                       */
00370         DCL ERR1 CHAR(80) INIT(
00380             ' ERROR:  PSU IDENTIFIER(S) NOT SPECIFIED.'),
00390             ERR2 CHAR(80) INIT(
00400             ' ERROR:  ADJ BY LIST NOT COMPATIBLE WITH PSU ID LIST.
    '),
00410             ERR3 CHAR(80) INIT(
00420             ' ERROR:  NO WEIGHT VARIABLES SPECIFIED.'),
00430             ERR4 CHAR(80) INIT(
00440             ' ERROR:  NUMPSU PARM MISSING OR INVALID.'),
00450             ERR5 CHAR(80) INIT(
00460             ' ERROR:  MORE THAN 64 REPLICATES SPECIFIED.'),
00470             ERR6 CHAR(80) INIT(
00480             ' ERROR:   STRATUM IDENTIFIER(S) NOT SPECIFIED.'),
00490             ERR7 CHAR(80) INIT(
00500             ' ERROR:   INVALID SPECIFICATION FOR NDEP.'),
00510             ERR1F FIXED BIN(31) BASED(ADDR(ERR1)),
00520             ERR2F FIXED BIN(31) BASED(ADDR(ERR2)),
00530             ERR3F FIXED BIN(31) BASED(ADDR(ERR3)),
00540             ERR4F FIXED BIN(31) BASED(ADDR(ERR4)),
00550             ERR5F FIXED BIN(31) BASED(ADDR(ERR5)),
00560             ERR6F FIXED BIN(31) BASED(ADDR(ERR6)),
00570             ERR7F FIXED BIN(31) BASED(ADDR(ERR7)):
00580         /*                                                       */
```

```
00590          /*   DISCOVER PROCEDURE OPTIONS..                          */
00600          /*                                                         */
00610          IF IOPT(1)=1 THEN MEANFLG='1'B :
00620          ELSE MEANFLG='0'B ;
00630          IF IOPT(2)=1 THEN STDFLG='1'B :
00640          ELSE STDFLG='0'B ;
00650          IF IOPT(3)=1 THEN COVFLG='1'B :
00660          ELSE COVFLG='0'B ;
00670          IF IOPT(4)=1 THEN CORRFLG='1'B :
00680          ELSE CORRFLG='0'B ;
00690          IF IOPT(5)=1 THEN REGRFLG='1'B :
00700          ELSE REGRFLG='0'B :
00710          IF IOPT(6)=1 THEN SUMFLG='1'B:
00720          ELSE SUMFLG='0'B:
00730          IF IOPT(7)=1 THEN DESFLG='1'B:
00740          ELSE DESFLG='0'B:
00750          IF IOPT(8)=1 THEN DUMFLG='1'B ;
00760          ELSE DUMFLG='0'B :
00770          IF IOPT(9)=1 THEN NOMSFLG='1'B:
00780          ELSE NOMSFLG='0'B:
00790          /*                                                         */
00800          /*   INSURE COMPATIBILITY OF OPTIONS & SET DEFAULTS        */
00810          /*                                                         */
00820          IF \(MEANFLG^STDFLG^COVFLG^CORRFLG^REGRFLG)
00830             THEN SUMFLG='1'B:
00840          IF COVFLG^CORRFLG^REGRFLG THEN SSQFLG='1'B:
00850                                    ELSE SSQFLG='0'B:
00860          IF SSQFLG THEN NOMSFLG='1'B:
00870          /*   DETERMINE NUMBER OF VARS ON EACH LIST                */
00880          NPSUID=NOVAR(3):
00890          IF NPSUID < 1 THEN DO:
00900
00910             CALL ERROR(ERR1F.ESTAB):
00920          END:
00930          NADJID=NOVAR(4):
00940          IF NADJID > NPSUID ^ NADJID > 0
00950             THEN DO:              /* RATIO ADJUSTMENTS */
00960                                   /* NOT YET HERE      */
00970               CALL ERROR(ERR2F.ESTAB):
00980             END:
00990             NSTRID=NOVAR(5):
01000             IF NSTRID < 1 THEN DO:
01010                CALL ERROR(ERR6F.ESTAB):
01020             END:
01030          /*                                                         */
01040          /*   DETERMINE NUMBER VARS ON WEIGHT STATEMENT..           */
01050          /*                                                         */
01060          IF   NOVAR(6) \= 1 THEN DO :
01070             CALL ERROR(ERR3F.ESTAB) :
01080          END :
01090          /* DETERMINE NO OF COMPUTED VARS                          */
01100          NCOMP=0:
01110          NCOMP=NOVAR(7):
01120
01130          /*                                                         */
01140          /*                    SET    NUMPSU                       */
01150          /*                                                         */
01160          NUMPSU=0.00:
01170          NUMPSU=PARM(1):
01180          /*                                                         */
```

```
01190          /*                          CHECK     NREP                    */
01200          /*                                                            */
01210          NREP=0.0;
01220          NREP=PARM(2);
01230          IF NREP < 1 THEN NREP=(NUMPSU/2+1);
01240             ELSE NREP=NREP+1;
01250          NV1=NOVAR(1)+1;
01260          /*                          CHECK     NDEP                    */
01270          NDEP=0;
01280          NDEP=PARM(3);
01290          IF REGRFLG THEN IF NDEP < 1 THEN NDEP=1;
01300          IF NDEP >= NOVAR(1) ^ (\REGRFLG & NDEP > 0) THEN DO;
01310             CALL ERROR(ERR7F,ESTAB);
01320          END;
01330    END SETPARM ;
01340   /*MEM3*/
01350 /*
01360 //LKED.SYSLMOD DD DSN=NCES.XEJGNY.SAGELIB.DISP=OLD
01370 //LKED.SYSLIB DD
01380 //             DD
01390 //             DD
01400 //             DD
01410 //             DD
01420 //             DD DSN=NCES.XEJGNY.SAGELIB.DISP=SHR
01430 //             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
01440 //             DD DSN=SYS2.SAS.R795.GLIBRARY.DISP=SHR
01450 //             DD DSN=SYS2.PLIXBASE,DISP=SHR
01460 //LKED.SYSIN  DD *
01470  NAME SETPARM(R)
01480 /*
END OF DATA
```

```
00010 //X         JOB (ED.AIR),'D-WYOUNG',PRTY=8,
00020 //             MSGCLASS=P,NOTIFY=XEJQNY,MSGLEVEL=(1,1)
00030 /*ROUTE PRINT R207
00040 //WYYSTEP1 EXEC XPLIXCL,SYSOUT=P,
00050 //    PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,OF,C',
00060 //    PARM.LKED='MAP,XREF,LIST,NCAL',
00070 //    COND.LKED=(4,LT,PLI)
00080 //PLI.SYSIN  DD *
00090   PROCESS:PROC(ESTAB) :
00100           /*-------------------------------------------------------*/
00110           /*                                                       */
00120           /*   PROCESS-COMPUTES ESTIMATES OF CHARACTERISTICS       */
00130           /*           SPECIFIED FOR EACH REPLICATE IN THE         */
00140           /*           SAMPLE DESIGN..                             */
00150           /*                                                       */
00160           /*           WRITES REPLICATE LEVEL RECORDS TO OUTPUT    */
00170           /*           DS IF THE OUTPUT OPTION IS IN EFFECT..      */
00180           /*                                                       */
00190           /*           COMPILES AND EXECUTES PROGRAMMING           */
00200           /*           STATEMENTS FOLLOWING THE PROC IF            */
00210           /*           THE OUTVAR/COMPVAR CONSTRUCTION             */
00220           /*           IS USED..                                   */
00230           /*                                                       */
00240           /*           NATIONAL ESTIMATES ARE STORED IN            */
00250           /*           ARRAY ESTAB(NREPS,VARNO)..                  */
00260           /*                                                       */
00270           /*-------------------------------------------------------*/
00280           /*                                                       */
00290           /*   EXTERNALS..                                         */
00300           /*                                                       */
00310           DCL (OUTFLG,OUTVFLG,SSQFLG,NOMSFLG,DESFLG,DUMFLG)
00320               BIT(1) EXTERNAL.
00330               (OUTPTR,INPTR,STRPTR,STRPTR2,
00340                PSUPTR,PSUPTR2) PTR EXTERNAL,
00350               (NOBS,NV1,NSTAT,NREP,NPSUID,NMISS,NSTRID)
00360               FIXED BIN(31) EXTERNAL,
00370                (WGTOBS, NUMPSU,WGTMISS) FLOAT BIN(53) EXTERNAL :
00380           /*                                                       */
00390           /*   SAS LINKAGE..                                       */
00400           /*                                                       */
00410           DCL INPUT ENTRY RETURNS(FIXED BIN(31)),
00420               VARX ENTRY(FIXED BIN(31),FLOAT BIN(53)),
00430               OBSPTR ENTRY RETURNS(PTR),
00440               SETDSN ENTRY(FIXED BIN(31)),
00450               PUTOUT ENTRY(FIXED BIN(31),PTR),
00460               FBUF ENTRY,
00470               BYPASS ENTRY(FIXED BIN(31)),
00480               CLSOUT ENTRY,
00490               IOPT  ENTRY(FIXED BIN(31))
00500                     RETURNS(FIXED BIN(31)),
00510               LINSIZ ENTRY(FIXED BIN(31)),
00520               STITLE ENTRY(FIXED BIN(31),FIXED BIN(31)),
00530               NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
00540                           FIXED BIN(15),),
00550               ERROR ENTRY(FIXED BIN(31),(*,*)FLOAT BIN(53)),
00560               MISSING ENTRY(FLOAT BIN(53)) RETURNS(BIT(1)):
00570           /*                                                       */
00580           /*  LOCALS..                                             */
00590           /*                                                       */
```

```
00600          DCL ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
00610              WGT FLOAT BIN(53) STATIC,
00620              NPSU FLOAT BIN(53) STATIC INIT(0),
00630              INHOLD(1) FLOAT BIN(53) BASED(INPTR),
00640              BYPSU(1)  FLOAT BIN(53) BASED(PSUPTR),
00650              BYPSU2(1) FLOAT BIN(53) BASED(PSUPTR2),
00660              BYCHPSU(1)  CHAR(8)     BASED(PSUPTR),
00670              BYCHPS2(1)  CHAR(8)     BASED(PSUPTR2),
00680              BYSTR(1) FLOAT BIN (53) BASED(STRPTR),
00690              BYSTR2(1) FLOAT BIN(53) BASED(STRPTR2),
00700              BYCHAR2(64) BIT(1) STATIC,
00710              BYCHSTR(1) CHAR(8) BASED(STRPTR),
00720              BYCHST2(1) CHAR(8) BASED(STRPTR2),
00730              BYCHAR(64)  BIT(1) STATIC,
00740              OUTVEC FIXED BIN(31) BASED(OUTPTR),
00750             (OPTR,IPTR) PTR,
00760             (ADDR,SUBSTR) BUILTIN,
00770             (IR,IV,K,NV,KV,IR1,JV,L,LNSZ,CENTER) FIXED BIN(31) ST
ATIC:
00780          DCL DESIGN(16)  CHAR(8)  STATIC INIT(
00790              /* 8*/   '1110100X',
00800              /*16*/   '11110101', '1001000X',
00810              /*24*/   '11111010', '11001100', '1010000X',
00820              /*32*/   '00001010', '11101100', '01111100', '1101001
X',
00830              /*48*/   '11111011', '11001010', '11100100', '1101100
0',
00840                       '10101100', '0010000X' );
00850          DCL MXREPS(5)   FIXED BIN(31) STATIC INIT(8, 16, 24, 32,
 48),
00860              REPIND(5)   FIXED BIN(31) STATIC INIT(1,  2,  4,  7,
 11),
00870              (X, WT, XWT,REPLID) FLOAT BIN(53) STATIC,
00880              DESVEC      CHAR(64)  BASED(DESPTR),
00890              (STRATNO, ISINDX, IRINDX) FIXED BIN(31) STATIC,
00900              (LASTPSU, TESTPSU)        CHAR(1)       STATIC,
00910              DESERR BIT(1) STATIC,
00920              DESPTR    POINTER,
00930              FLAT_EST(1) FLOAT BIN(53),
00940              1 NAMESTR,
00950              2 NTYPE    FIXED BIN(15),
00960              2 NPOS     FIXED BIN(15),
00970              2 NLNG     FIXED BIN(15),
00980              2 NVARO    FIXED BIN(15),
00990              2 NNAME    CHAR(8),
01000              2 NLABEL   CHAR(40),
01010              2 NFORM    CHAR(8),
01020              2 NIFORM   CHAR(8),
01030              2 NFL      FIXED BIN(15),
01040              2 NFD      FIXED BIN(15),
01050              2 NF       FIXED BIN(15),
01060              2 NJUST    FIXED BIN(15) :
01070          /*                                                    */
01080          /*   ERROR MESSAGES..                                 */
01090          /*                                                    */
01100          DCL ERR1 CHAR(80) INIT(
01110              ' ERROR:  MISSING VALUE FOR WEIGHT INVALID.'),
01120              ERR1F FIXED BIN(31) BASED(ADDR(ERR1)):
01130          DCL ERR2  CHAR(80) STATIC INIT(
01140              ' ERROR:  FILE NOT SORTED BY STRAT & PSU ID VARIABLE
```

```
S.').
01150              ERR2F FIXED BIN(31) BASED(ADDR(ERR2)):
01160           ON ENDPAGE(SYSPRINT) BEGIN :
01170              CALL STITLE(0,L) :
01180              PUT EDIT('PROC SRRVAR - BRR SAMPLING ERROR ESTIMATES'
)
01190                       (SKIP(1),COL(CENTER),A) :
01200              PUT SKIP (2) ;
01210           END:
01220
01230           /*  SET LINESIZE AND CENTER                        */
01240           /*                                                 */
01250           CALL LINSIZ(LNSZ):
01260           CENTER=(LNSZ-35)/2:
01270           IF IOPT(50) = 0 THEN CENTER=1:
01280
01290           /*   IF OUTPUT SPECIFIED DEFINE OUTPUT DS & VARS..  */
01300           /*                                                 */
01310           IF OUTFLG THEN CALL PREPOUT :
01320              /*                                             */
01330              /*    INITIALIZE VALUES                        */
01340              /*                                             */
01350               DO IV=1 TO NPSUID:
01360                  CALL NAMEV(3,IV,NTYPE,NAMESTR):
01370                  IF NTYPE=2 THEN
01380                     BYCHAR(IV)='1'B:
01390                  ELSE BYCHAR(IV)='0'B:
01400               END;
01410              DO IV=1 TO NSTRID:
01420                 CALL NAMEV(5,IV,NTYPE,NAMESTR):
01430                 IF NTYPE=2 THEN BYCHAR2(IV)='1'B:
01440                            ELSE BYCHAR2(IV)='0'B:
01450              END:
01460              LASTPSU='1'B:
01470              NOBS=0; WGTOBS=0:
01480              DO IR=1 TO NREP+1 :
01490                 DO IV=1 TO NSTAT:
01500                    ESTAB(IR,IV)=0.0 :
01510                 END ;
01520              END :
01530              NV=NV1-1:
01540              IR1=NREP+1:
01550           STRATNO=0;
01560           LASTPSU='1':
01570           DO IR = 1 TO 5:
01580              IF NREP <= MXREPS(IR)+1 THEN GO TO SETDES:
01590           END;
01600           IR=5:
01610   SETDES:
01620           IV=REPIND(IR):
01630           DESPTR=ADDR(DESIGN(IV)):
01640           ISINDX=1:
01650           IF DESFLG THEN DO:
01660              SIGNAL ENDPAGE(SYSPRINT):
01670              PUT EDIT('HALF SAMPLE DESIGN MATRIX','PSU','MATRIX')
01680                       (COL(CENTER),A,SKIP(2),A,COL(20),A) :
01690           END:
01700           PUT SKIP;
01710
01720              /*                                             */
```

```
01730              /*   STEP THRU OBS & COMPUTE ESTIMATES AT THE       */
01740              /*   REPLICATE LEVEL...                             */
01750              /*                                                  */
01760              DO WHILE(INPUT=0) :
01770                 CALL VARX(5, BYSTR(1)):
01780                 CALL VARX(3, BYPSU(1)):
01790
01800                 IF NOBS = 0 THEN GO TO NEWSTR:
01810              /*                    CHECK FOR NEXT STRATUM        */
01820              DO IV=1 TO NSTRID:
01830                 IF BYCHAR2(IV) THEN DO:
01840                    IF BYCHSTR(IV) > BYCHST2(IV)
01850                       THEN GO TO NEWSTR:
01860                       ELSE IF BYCHSTR(IV) < BYCHST2(IV)
01870                            THEN CALL ERROR(ERR2F,ESTAB):
01880                     END:
01890                     ELSE DO:
01900                       IF BYSTR(IV) > BYSTR2(IV)
01910                          THEN GO TO NEWSTR:
01920                          ELSE IF BYSTR(IV) < BYSTR2(IV)
01930                               THEN CALL ERROR(ERR2F,ESTAB):
01940                      END:
01950                   END:
01960              /*                     CHECK FOR NEXT PSU           */
01970                 DO IV = 1 TO NPSUID:
01980                   IF BYCHAR(IV) THEN DO:
01990                      IF BYCHPSU(IV) > BYCHPS2(IV) THEN GO TO NEWPS
U:
02000            ELSE IF BYCHPSU(IV)<BYCHPS2(IV) THEN CALL ERROR(ERR2F,E
STAB):
02010                   END:
02020                   ELSE IF BYPSU(IV) > BYPSU2(IV) THEN GO TO NEWPSU
:
02030              ELSE IF BYPSU(IV)<BYPSU2(IV) THEN CALL ERROR(ERR2F,ES
TAB):
02040                 END:
02050         GO TO PROCOBS;
02060  NEWSTR:
02070              IF NOBS > 0 THEN CALL ADDSUMS:
02080              IF LASTPSU='0' THEN DO;
02090                 CALL PRNTSTR:
02100                 PUT EDIT(' DUMMY PSU CREATED ')(A):
02110              END;
02120              STRATNO=STRATNO+1:
02130              ISINDX=ISINDX+1:
02140              IF SUBSTR(DESVEC,ISINDX,1)='X' THEN ISINDX=1:
02150              DESERR='0'B:
02160              LASTPSU='0':
02170              DO IV=1 TO NSTRID:
02180                 BYSTR2(IV)=BYSTR(IV):
02190              END;
02200              GO TO SETPSU:
02210  NEWPSU:
02220              CALL ADDSUMS:
02230              IF LASTPSU='0' THEN LASTPSU='1':
02240              ELSE DESERR='1'B:
02250  SETPSU:
02260              NPSU=NPSU+1:
02270              DO IV=1 TO NPSUID:
02280                 BYPSU2(IV)=BYPSU(IV):
```

```
02290                END:
02300                IF DESFLG^DESERR THEN DO:
02310                   CALL PRNTSTR:
02320                   CALL PRNTPSU:
02330                END:
02340                IF DESERR THEN
02350                   PUT EDIT(' MORE THAN 2 PSUS. THIS ONE SKIPPED')(A)
:
02360  PROCOBS:
02370                   CALL VARX(6,WGT):
02380                   IF MISSING(WGT) THEN DO:
02390                      CALL ERROR(ERRIF,ESTAB):
02400                   END;
02410                   NOBS=NOBS+1:  WGTOBS=WGTOBS+WGT:
02420                IF DESERR THEN GO TO ENDCASE;
02430                   CALL VARX(1,INHOLD(1)):
02440                   IF NOMSFLG THEN DO IV=1 TO NV;
02450                      IF MISSING(INHOLD(IV)) THEN DO:
02460                         NMISS=NMISS+1:
02470                         WGTMISS=WGTMISS+WGT:
02480                         GO TO ENDCASE:
02490                      END:
02500                   END;
02510
02520                      IR=NREP+1:
02530                      KV=0:
02540                      IF \NOMSFLG THEN KV=2*NV1:
02550                      K=2*NV1+KV:
02560                      DO IV=1 TO NV1;
02570                         IF IV = NV1 THEN X=1.0:  ELSE X=INHOLD(IV):
02580                         KV=KV+1:
02590                         IF \NOMSFLG THEN DO:
02600                            IF MISSING(X) THEN GO TO ENDVAR:
02610                            ESTAB(IR,IV)=ESTAB(IR,IV)+1.0:
02620                            ESTAB(IR,IV+NV1)=ESTAB(IR,IV+NV1)+WGT:
02630                         END;
02640                         XWT=X*WGT:
02650                         ESTAB(IR,KV)=ESTAB(IR,KV)+XWT:
02660                         ESTAB(IR,KV+NV1)=ESTAB(IR,KV+NV1)+X*XWT:
02670                         IF IV > 1 & SSQFLG = '1'B THEN DO JV=1 TO I
V-1:
02680                            K=K+1:
02690                            ESTAB(IR,K)=ESTAB(IR,K)+XWT*INHOLD(JV):
02700                         END:
02710  ENDVAR:              END;
02720  ENDCASE:     END:
02730
02740           CALL ADDSUMS:
02750           PUT SKIP EDIT(NUMPSU,' PSUS SPECIFIED',
02760                         NPSU,' PSUS FOUND')(F(10),A,SKIP,F(10),A):
02770           NUMPSU=NPSU;
02780           /*                                                      */
02790           /*   IF OUTPUT OPT IN EFFECT CLOSE THE DS..             */
02800           /*                                                      */
02810           IF OUTFLG THEN DO :
02820              CALL SETDSN(2) :
02830              CALL FBUF :
02840              CALL CLSOUT :
02850           END ;
02860           /*                                                      */
```

```
02870          /*                                                    */
02880       PREPOUT:PROC :
02890             /*-------------------------------------------------*/
02900             /*                                                 */
02910             /*   PREPOUT-INTERNAL PROC TO DEFINE OUTPUT DS     */
02920             /*           & ALL OUTPUT VARIABLES..              */
02930             /*           OUTPUT DS WILL CONTAIN THE            */
02940             /*           FOLLOWING:                            */
02950             /*                                                 */
02960             /*           1. REPLICATE ID    REPLID             */
02970             /*           2. ANY BY VARS     DEF ON INPUT       */
02980             /*           3. REPLICATE ESTS  DEF ON INPUT       */
02990             /*                                                 */
03000             /*-------------------------------------------------*/
03010             /*                                                 */
03020             /*   VAR DEF STRUCT..                              */
03030             /*                                                 */
03040             DCL 1 NAMESTR,
03050                 2 NTYPE   FIXED BIN(15),
03060                 2 NPOS    FIXED BIN(15),
03070                 2 NLNG    FIXED BIN(15),
03080                 2 NVARO   FIXED BIN(15),
03090                 2 NNAME   CHAR(8),
03100                 2 NLABEL  CHAR(40),
03110                 2 NFORM   CHAR(8),
03120                 2 NIFORM  CHAR(8),
03130                 2 NFL     FIXED BIN(15),
03140                 2 NFD     FIXED BIN(15),
03150                 2 NF      FIXED BIN(15),
03160                 2 NJUST   FIXED BIN(15) :
03170             /*                                                 */
03180             /*   SAS LINKAGE..                                 */
03190             /*                                                 */
03200             DCL OPNOUT ENTRY,
03210                 GETDS ENTRY(PTR),
03220                 HISTRY ENTRY(PTR),
03230                 NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
03240                             FIXED BIN(15),),
03250                 ONAME2 ENTRY(FIXED BIN(15),FIXED BIN(31)),
03260                 ONAMES ENTRY(FIXED BIN(15),FIXED BIN(31),
03270                              FLOAT BIN(53)),
03280                 ENDNAM ENTRY,
03290                 DLTOUT ENTRY,
03300                 ABUF ENTRY ;
03310             /*                                                 */
03320             /*  OUTVAR STRUCTURE...                            */
03330             /*                                                 */
03340             DCL 1 OUTSTR(1) BASED(OUTPTR),
03350                 2 OUTADDR  PTR,
03360                 2 VARADDR  PTR,
03370                 2 OUTNAME  CHAR(8) :
03380           /*                                                   */
03390           /*   DEFINE OUTPUT DS..                              */
03400           /*                                                   */
03410           CALL SETDSN(2) :          /* OUTPUT IS CURRENT.      */
03420           CALL OPNOUT :             /* OPEN OUTPUT DS..        */
03430           CALL SETDSN(1) :          /* INPUT IS CURRENT..      */
03440           CALL FBUF :               /*  FREE INPUT BUFFER      */
03450           CALL GETDS(IPTR) :        /*  ADDRESS INPUT..        */
03460           CALL SETDSN(2) :          /*  OUTPUT IS CURRENT      */
```

```
03470              CALL HISTRY(IPTR) :          /* GET DS HISTORY..     */
03480              /*                                                   */
03490              /*   DEFINE REPLICATE IDENTIFIER..                   */
03500              /*                                                   */
03510              NNAME='REPL_ID' :
03520              NLABEL='REPLICATE IDENTIFIER' :
03530              NFORM,NIFORM=' ' :
03540              NTYPE=1 :
03550              NLNG=8 :
03560              NFL,NFD= 0 :
03570              CALL ONAMES(NTYPE,OUTVEC,REPLID) :
03580              /*                                                   */
03590              /*   DEFINE VARS ON VARLIST..                        */
03600              /*                                                   */
03610              DO IV=1 TO NV1 :
03620                      CALL SETDSN(1) :
03630                      CALL NAMEV(1,IV,NTYPE,NAMESTR) :
03640                      NLNG=8 :
03650                      CALL SETDSN(2) :
03660                      CALL ONAMES(NTYPE,OUTVEC,FLAT_EST(IV)) :
03670              END :
03680              /*                                                   */
03690              /*   DEFINE COMPUTED VARS..                          */
03700              /*                                                   */
03710              IF OUTVFLG THEN DO IV=1 TO NSTAT:
03720                      NNAME=OUTNAME(IV) :
03730                      NLABEL='COMPUTED ESTIMATE' :
03740                      NLNG=8 :
03750                      NTYPE=1 :
03760                      NFL,NFD=0 :
03770                      NFORM,NFORM=' ' :
03780                      CALL ONAMES(NTYPE,OUTVEC,FLAT_EST(IV+NV1))
 :
03790              END :
03800              /*                                                   */
03810              /*   TERMINATE DS & OUTPUT VAR DEFINITION..          */
03820              /*   DELETE OUTPUT ROUTINES...                       */
03830              /*                                                   */
03840              CALL ENDNAM :
03850              CALL DLTOUT :
03860              CALL ABUF :
03870              CALL SETDSN(1) :
03880              CALL ABUF :
03890      END PREPOUT :
03900 ADDSUMS: PROC:
03910      /*     INTERNAL PROC TO ADD PSU SUMS      */
03920      /*     TO HALF-SAMPLE SUMS                */
03930              IR1=NREP+1:
03940              WGT=1.0:
03950                 IRINDX=ISINDX:
03960                 DO IR=1 TO NREP:
03970                    IF IR > 2 THEN DO:
03980                       IRINDX=IRINDX+1:
03990                       TESTPSU=SUBSTR(DESVEC,IRINDX,1):
04000                       IF TESTPSU = 'X' THEN DO:
04010                          IRINDX=1:  TESTPSU=SUBSTR(DESVEC,1,1):
04020                       END:
04030                       IF DESFLG THEN IF LASTPSU=TESTPSU
04040                          THEN PUT EDIT('0 ')(A):
04050                          ELSE PUT EDIT('1 ')(A):
```

```
04060                         IF LASTPSU=TESTPSU THEN GO TO ENDREP;
04070                       END;
04080                       /*  SECOND REP INCLUDES LAST OF EACH PAIR */
04090                       ELSE IF IR = 2 THEN DO:
04100                         IF DESFLG THEN PUT EDIT(LASTPSU,' ')(A,A);
04110                         WGT=2.0*WGT;  /* DOUBLE WEIGHT FOR HALF SAMP
LES */
04120                         IF LASTPSU = '0' THEN GO TO ENDREP;
04130                        END;
04140                        DO IV=1 TO NSTAT;
04150                           ESTAB(IR,IV)=ESTAB(IR,IV)+
04160                                        WGT*ESTAB(IR1,IV);
04170                        END;
04180  ENDREP:          END;
04190      DO IV=1 TO NSTAT;
04200         ESTAB(IR1,IV)=0.0;
04210      END;
04220  END ADDSUMS;
04230  PRNTSTR: PROC;
04240       /*      INTERNAL PROC TO PRINT STRATUN ID       */
04250               PUT SKIP;
04260               DO IV=1 TO NSTRID;
04270                  IF BYCHAR2(IV) THEN
04280                     PUT EDIT (BYCHST2(IV),' ')(A,A);
04290                     ELSE PUT EDIT (BYSTR2(IV),' ')(F(8),A);
04300               END;
04310  END PRNTSTR;
04320
04330  PRNTPSU: PROC;
04340       /*      INTERNAL PROC TO PRINT PSU ID      */
04350               DO IV=1 TO NPSUID;
04360                  IF BYCHAR(IV)
04370                     THEN PUT EDIT(BYCHPS2(IV),' ')(A,A);
04380                     ELSE PUT EDIT(BYPSU2(IV),' ')(F(8),A);
04390               END;
04400  END PRNTPSU;
04410   END PROCESS ;
04420  /*MEM6*/
04430 /*
04440 //LKED.SYSLMOD DD DSN=NCES.XEJGNY.SAGELIB.DISP=SHR
04450 //LKED.SYSLIB DD
04460 //             DD
04470 //             DD
04480 //             DD
04490 //             DD
04500 //             DD DSN=NCES.XEJGNY.SAGELIB.DISP=SHR
04510 //             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
04520 //             DD DSN=SYS2.SAS.R795.GLIBRARY.DISP=SHR
04530 //LKED.SYSIN  DD *
04540  NAME PROCESS(R)
04550 /*
END OF DATA
```

```
00010 //X         JOB (ED,AIR),'D-WYCUNG',PRTY=8,
00020 //              MSGCLASS=P,NOTIFY=XEJGNY,MSGLEVEL=(1,1)
00030 /*ROUTE  PRINT R207
00040 //WYYSTEP1 EXEC XPLIXCL,SYSOUT=P,
00050 //    PARM.PLI='NM,ND,A,NEST,X,NC(S),MAR(2,72,1),NSYN(C),NOP,OF,C',
00060 //    PARM.LKED='MAP,XREF,LIST,NCAL',
00070 //    COND.LKED=(4,LT,PLI)
00080 //PLI.SYSIN  DD *
00090   PRINTIT:PROC(ESTAB) :
00100           /*                                                    */
00110           /*  EXTERNALS..                                       */
00120           /*                                                    */
00130           DCL (NV1,NOUT,NWGT,NOBS,NREP,NDEP,NMISS,NSTAT)
00140               FIXED BIN(31) EXTERNAL,
00150               (NUMPSU,WGTOBS,WGTMISS) FLOAT BIN(53) EXTERNAL,
00160               (MISSPTR,MWGTPTR) PTR EXTERNAL,
00170               (SUMFLG,MEANFLG,STDFLG,SSQFLG,COVFLG,CORRFLG,REGRFLG
                    ,
00180                NOMSFLG,DUMFLG) BIT(1) EXTERNAL :
00190           /*                                                    */
00200           /*  INTERNAL STRUCTURES..                             */
00210           /*                                                    */
00220           DCL ESTAB(*,*) FLOAT BIN(53) CONTROLLED,
00230               1 NAMESTR,
00240               2 NTYPE    FIXED BIN(15),
00250               2 NPOS     FIXED BIN(15),
00260               2 NLNG     FIXED BIN(15),
00270               2 NVARO    FIXED BIN(15),
00280               2 NNAME    CHAR(8),
00290               2 NLABEL   CHAR(40),
00300               2 NFORM    CHAR(8),
00310               2 NIFORM   CHAR(8),
00320               2 NFL      FIXED BIN(15),
00330               2 NFD      FIXED BIN(15),
00340               2 NF       FIXED BIN(15),
00350               2 NJUST    FIXED BIN(15) :
00360            DCL SYSPRINT FILE PRINT;
00370           /*                                                    */
00380           /*   DECLARE LINKAGE TO SAS...                        */
00390           /*                                                    */
00400           DCL NAMEV ENTRY(FIXED BIN(31),FIXED BIN(31),
00410                           FIXED BIN(15),),
00420               BEST  ENTRY(FLOAT BIN(53),FIXED BIN(31),
00430                           FLOAT BIN(53)),
00440               IOPT  ENTRY(FIXED BIN(31))
00450                     RETURNS(FIXED BIN(31)),
00460               LINSIZ ENTRY(FIXED BIN(31)),
00470               STITLE ENTRY(FIXED BIN(31),FIXED BIN(31)),
00480               ERROR ENTRY(FIXED BIN(31),(*,*)FLOAT BIN(53)) :
00490           /*                                                    */
00500           /*  LOCALS..                                          */
00510           /*                                                    */
00520           DCL (I,IR,IV,JV,K,NV,L,CENTER,LNSZ,IER,KV,KV1,II,NP,NP1,
IR1)
00530               FIXED BIN(31),
00540                (ADDR,SUBSTR,SQRT) BUILTIN,
00550                (EST,SE,LB,UB,XNR,XM,XN,XS) FLOAT BIN(53),
00560                FCHAR CHAR(10) ALIGNED,
00570                DBEST FLOAT BIN(53) BASED(ADDR(FCHAR)),
```

```
00580              STATS(10) FLOAT BIN(53):
00590           /*                                                  */
00600           /*   ERROR MESSAGES..                               */
00610           /*                                                  */
00620           DCL SUBTIT  CHAR(32) STATIC INIT('SUMMARY INFORMATION'):
00630           DCL STATLBL CHAR(10) STATIC:
00640
00650           DCL ERR1 CHAR(80) INIT(
00660               ' ERROR:  LINE SIZE MUST BE > 132 IN PRESENT VERSION
                .').
00670               ERR1F FIXED BIN(31) BASED(ADDR(ERR1)):
00680           /*                                                  */
00690           /*   DEFINE ENDPAGE CONDITION.. '                   */
00700           /*                                                  */
00710           ON ENDPAGE(SYSPRINT) BEGIN :
00720              CALL STITLE(0,L) :
00730              PUT EDIT('PROC BRRVAR - BRR SAMPLING ERROR ESTIMATES'
                )
00740                       (SKIP(1).COL(CENTER).A) :
00750              PUT SKIP EDIT(SUBTIT)(COL(CENTER).A):
00760              PUT SKIP (2) :
00770              IF SUBSTR(SUBTIT.1.1) = 'U' THEN DO:
00780                 PUT EDIT('95% CONFIDENCE BOUNDS', 'STATISTIC',
00790                          'VARIABLE', 'WTD. N',
00800                     'ESTIMATE', 'STD. ERROR', 'LOWER', 'UPPER', 'L
                ABEL')
00810                      (COL(68).A.SKIP.A.COL(12).A.COL(26).A.COL(39).
                A.
00820                       COL(50).A.COL(67).A.COL(80).A.COL(87).A):
00830                END:
00840                 ELSE IF SUBSTR(SUBTIT.1.1) = 'C' THEN DO:
00850                    PUT EDIT('FIRST', 'SECOND', SUBSTR(SUBTIT.1.11).
00860                        '95% CONFIDENCE BOUNDS', 'VARIABLE', 'VARIABLE
                        '.
00870                        'WTD. N','ESTIMATE','STD. ERROR','LOWER ','UPP
                        ER'.
00880                        'PROB C>0')
00890                        (A. COL(12), A. COL(22), A, COL(63). A. SKIP.
00900                         A. COL(12). A, COL(26). A. COL(39). A. COL(51
                        ). A.
00910                         COL(64). A. COL(77). A.COL(87).A):
00920                    END:
00930                    ELSE IF SUBSTR(SUBTIT.1.1) = 'R' THEN DO:
00940                       PUT EDIT('FIRST', 'SECOND', SUBSTR(SUBTIT,1,11),
00950                           '95% CONFIDENCE BOUNDS', 'VARIABLE', 'VARIABLE
                           '.
00960                           'WTD. N','ESTIMATE','STD. ERROR','LOWER ','UPP
                           ER'.
00970                           'PROB C>0')
00980                           (A, COL(12), A, COL(22), A, COL(68). A. SKIP.
00990                            A, COL(12). A, COL(26), A, COL(39). A. COL(51
                           ), A.
01000                            COL(64). A. COL(77), A.COL(87).A):
01010                       END:
01020                       PUT SKIP:
01030           END:
01040
01050           /*  SET LINESIZE AND CENTER                         */
01060           CALL LINSIZ(LNSZ);
01070           IF LNSZ < 80 THEN CALL ERROR(ERR1F. ESTAB):
```

```
01080          CENTER=(LNSZ-35)/2:
01090           IF IOPT(50) = 0 THEN CENTER=1:
01100
01110          /*  PRINT SUMMARY REPORT                          */
01120           SIGNAL ENDPAGE(SYSPRINT):
01130           PUT SKIP EDIT(NOBS,' OBSERVATIONS READ')
01140               (F(12),A):
01150           PUT SKIP EDIT(WGTOBS,' WEIGHTED OBSERVATIONS READ')
01160               (F(12.1), A):
01170           PUT SKIP:
01180           IF NOMSFLG THEN DO:
01190              WGTOBS=WGTOBS-ESTAB(1,NV1):
01200              PUT SKIP EDIT(WGTOBS,
01210                  ' WEIGHTED OBS DELETED FOR MISSING VALUES')(F(12.
01220           END:
01230           ELSE PUT SKIP EDIT('   MISSING VALUES DELETED SEPARATELY
                    'FOR EACH VAR')(A,A):
01240           PUT SKIP:
01250           PUT SKIP(2) EDIT(NREP,' REPLICATES IN',NUMPSU,
01260               'PSU PAIR DESIGN')(F(12),X(1),A,F(4),X(1),A):
01270
01280          /*  COMPUTE AND PRINT SUMS                        */
01290          SUBTIT = 'UNIVARIATE STATISTICS':
01300          NV=NV1-1:  XNR=NREP-1:
01310          IF NOMSFLG THEN KV1=0:
01320             ELSE KV1=2*NV1:
01330          IF SUMFLG THEN DO:
01340             SIGNAL ENDPAGE(SYSPRINT):
01350             STATLBL='SUM':
01360             XN=ESTAB(1,NV1):
01370             KV=KV1:
01380             DO IV=1 TO NV:
01390                KV=KV+1:
01400                IF \NOMSFLG THEN XN=ESTAB(1,IV+NV1):
01410                CALL PRINTLN(KV,IV,XN):
01420                PUT EDIT(NLABEL)(COL(74),A):
01430             END:
01440          END:
01450
01460          /*  COMPUTE MEANS                                 */
01470          DO IR=1 TO NREP:
01480             KV=KV1:
01490             XN=ESTAB(IR, NV1):
01500             DO IV=1 TO NV:
01510                KV=KV+1:
01520                IF \NOMSFLG THEN XN=ESTAB(IR,IV+NV1):
01530                ESTAB(IR, KV)=ESTAB(IR,KV)/XN:
01540             END:
01550          END;
01560
01570          IF MEANFLG THEN DO:
01580             STATLBL='MEAN':
01590             XN=ESTAB(1,NV1):
01600             KV=KV1;
01610             DO IV=1 TO NV:
01620                KV=KV+1:
01630                IF \NOMSFLG THEN XN=ESTAB(1,IV+NV1):
01640                CALL PRINTLN(KV,IV,XN);
01650
```

```
01660                PUT EDIT(NLABEL)(COL(87),A);
01670             END;
01680          END;
01690
01700          /* COMPUTE S. D.S                                  */
01710          DO IR=1 TO NREP;
01720             XN=ESTAB(IR,NV1);
01730             KV=KV1;
01740             DO IV=1 TO NV;
01750                KV=KV+1;
01760                IF \NOMSFLG THEN XN=ESTAB(1,IV+NV1);
01770                SE=ESTAB(IR,KV+NV1)/XN-ESTAB(IR,KV)**2;
01780                IF SE > 0 THEN SE=SQRT(SE);
01790                ESTAB(IR,KV+NV1)=SE;
01800             END;
01810          END;
01820
01830          IF STDFLG THEN DO;
01840             STATLBL='STD';
01850             XN=ESTAB(1,NV1);
01860             KV=KV1;
01870             DO IV=1 TO NV;
01880                KV=KV+1;
01890                IF \NOMSFLG THEN XN=ESTAB(1,IV+NV1);
01900                CALL PRINTLN(KV+NV1,IV,XN);
01910                PUT EDIT(NLABEL)(COL(74),A);
01920             END;
01930          END;
01940
01950          /* COMPUTE COVARIANCES AND CORRELATIONS */
01960          IF SSQFLG & NV > 1
01970             THEN DO IR=1 TO NREP;
01980             K=2*NV1;
01990             DO IV=2 TO NV;
02000                XM=ESTAB(IR,IV);  XN=ESTAB(IR,NV1);
02010                DO JV=1 TO IV-1;
02020                   K=K+1;
02030                   ESTAB(IR,K)=(ESTAB(IR,K)/XN-ESTAB(IR,JV)*XM);
02040                END;
02050             END;
02060          END;
02070
02080          IF CORRFLG THEN DO;
02090             DO IR=1 TO NREP;
02100                K=2*NV1;
02110                DO IV=2 TO NV;
02120                   XS=ESTAB(IR,IV+NV1);
02130                   DO JV=1 TO IV-1;
02140                      K=K+1;
02150                      ESTAB(IR,K)=ESTAB(IR,K)
02160                           /(XS*ESTAB(IR,JV+NV1));
02170                   END;
02180                END;
02190             END;
02200          END;
02210
02220           IF COVFLG^CORRFLG THEN DO;
02230              IF CORRFLG THEN
02240                      SUBTIT='CORRELATION ESTIMATES';
02250                 ELSE SUBTIT='COVARIANCE ESTIMATES';
```

```
02260              XN=ESTAB(1,NV1):
02270           SIGNAL ENDPAGE(SYSPRINT):
02280           K=2*NV1:
02290              DO IV=2 TO NV:
02300                 CALL NAMEV(1,IV,NTYPE,NAMESTR):
02310                 STATLBL=NNAME;
02320                 DO JV=1 TO IV-1:
02330                    K=K+1:
02340                    CALL PRINTLN(K,JV,XN):
02350                 END:
02360              END;
02370        END:
02380
02390        DCL CPTR PTR,   K2 FIXED BIN(31),
02400            C(1) FLOAT BIN(53) BASED(CPTR):
02410        IF REGRFLG THEN DO:
02420           SUBTIT='REGRESSION COEFFICIENTS':
02430           SIGNAL ENDPAGE(SYSPRINT):
02440           IR1=NREP+1:
02450           CPTR=ADDR(ESTAB(IR1,1)):
02460           NP=NV1-NDEP-1:
02470           NP1=NP+1:
02480           DO II=1 TO NDEP:
02490              KV=NP+II:
02500              DO IR=1 TO NREP:
02510              K=2*NV1:
02520              K2=0:
02530                DO IV=1 TO NP:
02540                 IF IV > 1 THEN
02550                    DO JV=1 TO IV-1:
02560                       K=K+1:    K2=K2+1:
02570                       C(K2)=ESTAB(IR,K);
02580                    END:
02590                    K2=K2+1;
02600                    IF CORRFLG THEN C(K2)=1.0;
02610                       ELSE C(K2)=ESTAB(IR,IV+NV1)**2:
02620                 END:
02630                 K=(KV-1)*(KV-2)/2+2*NV1:
02640                 DO IV=1 TO NP:
02650                    K=K+1:     K2=K2+1:
02660                    C(K2)=ESTAB(IR,K):
02670                 END:
02680                 K2=K2+1;
02690                 IF CORRFLG THEN XS=1.0:
02700                    ELSE XS=ESTAB(IR,II+NP+NV1)**2:
02710                    C(K2)=XS:
02720                 CALL REGRLT(C,NP1,IER):
02730                 K=0:
02740                 K2=NP*(NP+1)/2:
02750                 DO IV=1 TO NP:
02760                    K=K+1:     K2=K2+1:
02770                    ESTAB(IR,K)=C(K2):
02780                 END:
02790                 K=K+1:     K2=K2+1:
02800                 ESTAB(IR,K)=(XS-C(K2))/XS:
02810              END:
02820                 CALL NAMEV(1,KV,NTYPE,NAMESTR):
02830                 STATLBL=NNAME:
02840                 DO IV=1 TO NP:
02850                    CALL PRINTLN(IV,IV,XN);
```

```
02860                END:
02870                 STATLBL='MULT RSQ':
02880                 CALL PRINTLN(NP1,KV,XN):
02890             END:
02900           END:
02910
02920         /*    PRINT ESTIMATES FOR EACH HALF SAMPLE                */
02930         /*                                                        */
02940         IF DUMFLG THEN DO:
02950         SUBTIT='REPLICATE STATISTICS':
02960         SIGNAL ENDPAGE(SYSPRINT):
02970        DO IV=1 TO NSTAT:
02980           PUT SKIP:
02990            DO IR=1 TO NREP:
03000               PUT EDIT(ESTAB(IR,IV))(F(8,3)):
03010            END;
03020           END:
03030         END:
03040
03050     PRINTLN: PROC(K,IV,XN):
03060           /*  SUBROUTINE TO SET UP AND PRINT A SINGLE LINE */
03070           /*  K POINTS TO THE ESTIMATE IN ESTAB            */
03080          DCL (K,IV)    FIXED BIN(31):
03090          DCL XN        FLOAT BIN(53):
03100
03110           CALL NAMEV(1,IV,NTYPE,NAMESTR):
03120           EST=ESTAB(1,K):
03130           CALL SAMPERR(ESTAB, K, XNR, SE):
03140           LB=EST-1.96*SE:   UB=EST+1.96*SE:
03150           PUT SKIP EDIT(STATLBL, NNAME, XN, EST, SE, LB, UB, NLABE
L)
03160               (A, COL(12), A, COL(20), F(13,1),F(13,3),F(13,3),
03170                F(13,3), F(13,3), A):
03180     END PRINTLN:
03190
03200     SAMPERR: PROC(ESTAB, II, XNR, SE):
03210           /*  INTERNAL SUBROUTINE TO COMPUTE THE SAMPLING STD ERRO
R*/
03220          DCL ESTAB(*, *)  FLOAT BIN(53)  CONTROLLED:
03230           DCL (SE, XNR)     FLOAT BIN(53),
03240               (IR, II)     FIXED BIN(31),
03250               NREP         FIXED BIN(31) EXTERNAL:
03260
03270          SE=0.0:
03280           EST=ESTAB(1,II);
03290           DO IR=2 TO NREP:
03300              SE=SE+(ESTAB(IR,II)-EST)**2:
03310           END:
03320           IF SE > 0 THEN SE=SQRT(SE/XNR) :
03330      END SAMPERR;
03340     REGRLT:  PROC(C,NV,IER):
03350        /*     REPLACES A LOWER TRIANGULAR COV MTX      */
03360        /*     WITH A MTX OF REGR COES WITH ERROR       */
03370        /*     VARS ON THE MAIN DIAG.  (EACH VAR IS     */
03380        /*     PREDICTED BY THE PRECEDING ONES)         */
03390        /*     IER IS SET TO THE NUMBER                 */
03400
03410        DCL C(1)      FLOAT BIN(53),
03420           (B,D,E)  FLOAT BIN(53) STATIC,
03430           (NV,IER) FIXED BIN(31),
```

```
03440           (N,K,I,KP,KR,KC,KE,JJ,J) FIXED BIN(31) STATIC:
03450
03460       IER=0:
03470       N=NV-1:
03480       KP=0:
03490       DO K=1 TO N:
03500          KR=KP:      KP=KP+K:
03510          D=C(KP):
03520          IF D < 1.E-20 THEN DO:
03530             IER=1;
03540             GO TO ENDREG:
03550          END;
03560          DO I=1 TO NV:
03570             IF I < K THEN DO:
03580                KR=KR+1:      KC=KP+K:      JJ=K+1:
03590             END:
03600             ELSE IF I > K THEN DO:
03610                KR=KR+I-1:      KC=KR:        JJ=I:
03620             END:
03630             ELSE DO:
03640                KR=KR+1:
03650                GO TO NEXTI:
03660             END:
03670             B=C(KR)/D:
03680             DO J=JJ TO NV:
03690                KE=KC+I-K:
03700                C(KE)=C(KE)-B*C(KC):
03710                KC=KC+J:
03720             END:
03730             IF I > K THEN C(KR)=B:
03740       NEXTI: END:
03750       END:
03760     ENDREG:
03770   END REGRLT:
03780   END PRINTIT:
03790   /* MEMB */
03800 /*
03810 //LKED.SYSLMOD DD DSN=NCES.XEJGNY.SAGELIB.DISP=OLD
03820 //LKED.SYSLIB DD DSN=NCES.XEJGNY.SAGELIB.DISP=SHR
03830 //             DD DSN=SYS2.SAS.V796.SUBLIB.DISP=SHR
03840 //             DD DSN=SYS2.PLIXBASE.DISP=SHR
03850 //             DD DSN=SYS2.PLIXLINK.DISP=SHR
03260 //LKED.SYSIN  DD *
03870   NAME PRINTIT(R)
03880 /*
END OF DATA
```

PARSER

```
00010 //X         JOB (ED.AIR),'D-WYOUNG',PRTY=8,
00020 //              MSGCLASS=P,NOTIFY=XEJONY,MSGLEVEL=(1,1)
00030 /*ROUTE PRINT   R93
00040 //ASM  EXEC XASMVCL,CREG=500K
00050 //ASM.SYSPRINT  DD SYSOUT=P
00060 //ASM.SYSLIB DD DSN=SYS2.SAS.V796.MACLIB,DISP=SHR
00070 //ASM.SYSIN DD *
00080            PRINT   NOGEN
00090 PROC       SASPROC NAME=BRRVAR,LOADMOD=BRRVAR2,DATA=ASGIVEN
00100 OPTS       SASLIST MEAN,1,MEANS,1
00110 OPTS       SASLIST STD,2
00120 OPTS       SASLIST COV,3
00130 OPTS       SASLIST COR,4,CORR,4
00140 OPTS       SASLIST REG,5,REGR,5
00150 OPTS       SASLIST SUM,6,SUMS,6
00160 OPTS       SASLIST DES,7,DESIGN,7
00170 OPTS       SASLIST DUMP,8
00180 OPTS       SASLIST NOMISS,9,NM,9
00190 PARMS      SASLIST NUMPSU,1,NPSU,1,MODE=NUMERIC
00200 PARMS      SASLIST NREP,2,NREPS,2,MODE=NUMERIC
00210 PARMS      SASLIST NDEP,3,NDEPS,3,MODE=NUMERIC
00220 LISTS      SASLIST VARIABLES,1,VAR,1,MODE=NUMERIC
00230 LISTS      SASLIST CLASSES,2,CLASS,2,MODE=ALPHANUM
00240 LISTS      SASLIST PSUID,3,PSU,3,MODE=ALPHANUM
00250 LISTS      SASLIST ADJBY,4,MODE=ALPHANUM
00260 LISTS      SASLIST STRATID,5,STR,5,MODE=ALPHANUM
00270 LISTS      SASLIST WEIGHT,6,WGT,6,MODE=NUMERIC
00280 LISTS      SASLIST COMPUTE,7,COMP,7,MODE=NUMERIC
00290 DATASETS   SASLIST DATA,1,MODE=DSIN
00300 DATASETS   SASLIST OUT,2,MODE=DSOUT
00310 DATASETS   SASLIST OUTEST,3,MODE=DSOUT
00320 DATASETS   SASLIST ADJDATA,4,MODE=DSIN
00330 DEFAULT    SASDFLT NO=1,NAME=_LAST_,MODE=DSIN
00340 DEFAULT    SASDFLT NO=2,NAME=_DATA_,MODE=DSOUT
00350            BR    RT
00360 INIT       ENTER
00370            SASLOAD SASART
00380            BALR    14,15
00390            STM     0,1,COMPLINK
00400            MVC     @NAME.=CL8'_OBS_'
00410            BAL     RT,PSUEDO
00420 INIT       LEAVE
00430 PSUEDO     MVI     @MAJNAME,C' '
00440            MVI     @SW,@SWPSE+@SWDEF
00450            MVI     @MODE,@A+@S
00460            L       15,$ADDLOC
00470            BR      15
00480 BRRVARD    DSECT
00490 COMPLINK   DS      2F
00500 BRRVAR     CSECT
00510 TERMER     LM      15,0,COMPLINK
00520            LA      1,1
00530            BR      15
00540 TERMSUB    ENTER
00550            LM      15,0,COMPLINK
00560            LA      1,2
00570            BALR    14,15
00580            SASDLT  SASART
00590 TERMSUB    LEAVE
```

```
00600 STMTREC  ENTER
00610          COMP     OUTVAR.OUTVAR
00620          COMP     OUTVARS.OUTVAR
00630          COMP     OUTV.OUTVAR
00640          COMP     COMPVAR.CMPVAR
00650          COMP     COMPV.CMPVAR
00660          COMP     COMP.CMPVAR
00670          LM       15.0.COMPLINK
00680          SR       1.1
00690          BALR     14.15
00700 STMTREC  LEAVE
00710 OUTVAR   LA       0.16
00720          BAL      RT.STMTST
00730          WORD     .
00740          SR       5.5
00750 OUTL     COMP     C':'.OUTE
00760          LIST     OUTI.OUTL
00770          NAME
00780          DATA     DATA=W1.N=8
00790          LA       5.1(5)
00800          B        OUTI
00810 OUTE     ST       5.SUBLIST
00820          BAL      RT.STMTEND
00830          B        STMTRECX
00840 CMPVAR   L        6.@DSADDR
00850          USING    D$TBL.6
00860          WORD     .
00870 CMPL     COMP     C':'.CMPE
00880          LIST     COMPI.CMPL
00890          LA       0.D$VAR
00900          ST       0.@LIST
00910          DO       $VARFIND
00920          LH       0.@VARNO
00930          LTR      0.0
00940          BZ       COMPI
00950          LPADD    D$VARLST+16
00960          MVC      LNAME(12).@NAME
00970          MVC      22(2,2).@VARNO
00980          MVI      @MODE.@A+@S
00990          MVI      @SW.@SWDEF
01000          MVI      @MAJNAME.C' '
01010          DO       $ADDLOC
01020          B        COMPI
01030 CMPE     B        STMTRECX
01040          SASEND   INIT=INIT.STMTREC=STMTREC.TERMERR=TERMER.TERM=TERMSUB
01050          LTORG
01060          END
01070 //LKED.SYSLMOD DD DSN=NCES.XEJONY.SAGELIB.DISP=SHR
01080 //LKED.SYSPRINT  DD SYSOUT=P
01090 //LKED.SYSIN DD *
01100  SETSSI AC000001
01110  NAME BRRVAR(R)
END OF DATA
```